Riding and thriving on the API hype cycle

Vukovic, Maja; Laredo, Jim; Muthusamy, Vinod; Slominski, Aleksander; Vaculin, Roman; ...
[+] Communications of the ACM , Volume 59 (3) – Feb 25, 201

Application programming interfaces (APIs) are, in the simplest term, specifications that govern interoperability between applications and services. Over the years, the API paradigm has evolved from beginnings as purpose-built initiatives to spanning entire application domains.8 Driven by the promise of new business opportunities, enterprises are increasingly investigating API ecosystems. In this Viewpoint, we discuss the challenges enterprises face in capitalizing on the potentials of API ecosystems. Is the investment in API ecosystems worth the promise of new profits? From a technical perspective, standardization of APIs and a systematic approach to consumability are critical for a successful foray into API ecosystems.

The Web API Economy: No Longer SOA's Adolescence

When the service-oriented architecture (SOA) concept emerged in the early 2000s, it attracted many companies that saw the benefits of bolstering business-to-business relationships via standard interfaces, often implemented via the simple object access protocol (SOAP). Later, SOA evolved into more Web-friendly technologies, such as REpresentational State Transfer (REST) that greatly simplified reusability, partly because the create, read, update, and delete (CRUD) interface was more approachable to even the most casual developers. While SOA was largely confined to the enterprise and focused on interoperability, REST APIs brought the power of reuse within reach of individual developers at Internet-scale through consumability.7 Developers now enjoyed cheap and easy access to deep computing capabilities and vast amounts of data that were hitherto hidden behind closed enterprises, driving today's surging API ecosystems. Today, organizations are heavily competing in the "API game," rapidly externalizing their business assets and hoping to vastly monetize locked up data and services.

ProgrammableWeb (see http://programmableweb.com), the largest free online API registry, claims API availability exhibited a compounded annual growth rate of approximately 100% from 2005 to 2011. As of April 2015, ProgrammableWeb listed over 13,300 APIs. This trend is not necessarily the result of altruistic nurturing of crowd-based creativity—APIs are huge business enablers! For example, Salesforce as a leading enterprise customer relationship management (CRM) provider, offers APIs to enable broad proliferation of its CRM capabilities into its customers' own systems. Today, 60% of Salesforce's transactions go through its API, contributing to its 1.3 billion daily transactions and more than $5 billion in annual revenue.8 APIs are primary business drivers not only for the born-on-the-cloud

businesses, but are also helping traditional businesses, such as financial services, to reinvent for their survival and prosperity.[5]

To extract value from some business asset, a set of services interconnected by APIs must be established. APIs that are part of an ecosystem are more valuable than when they exist in isolation. To create successful business models around the API economy, it is important to develop an ecosystem of partners and consumers. It is also important to understand technology trends affecting applications that consume APIs. Container-based technologies such as Docker[a] are making it possible for developers and service providers to rapidly develop and deploy their services using standardized interfaces. This allows the ecosystem to rapidly evolve while the participants focus on their core competencies. In such an ecosystem, Platform-as-a-Service (PaaS) providers manage the IT service automation using APIs, while Software-as-a-Service (SaaS) providers supply specialized services. Successful business models find a niche in an existing ecosystem or create new ecosystems using APIs.

The greatest risk for enterprises remains a lack of a sound API strategy.[5] We discuss key challenges that must be overcome to tame inflated API expectations and form a healthy, self-evolving API ecosystem.

API success factors. API ecosystems bring much more attention to API consumability, ease of reuse, and reuse in contexts not originally envisioned by the provider—sometimes referred to as serendipitous reuse.[10] APIs initially generated a lot of hype for enterprises, given the potential of new client bases through the promise of almost accidental reuse of APIs. Yet, the critical question is how can enterprises design the desirable APIs, for easy reuse, and avoid the investment loses from (re-)design of ineffective APIs and deployment of infrastructure to host and support them? The probability of API success is largely a function of where an organization is in its digital evolution.[4] On one hand, there exist born-on-the-Web businesses that have developed their core identities around APIs (for example, Twilio[b] and Stripe[c]). Such companies enjoyed the first-mover advantage and are currently benefitting from huge consumer demand. On the other hand, there are the pre-Web businesses (such as large banks and healthcare institutions), which have only recently started investigating API strategies. They typically have to proceed along the API path more cautiously since the market has become flooded and many consumers already have their preferred APIs. Another factor that seemingly affects API success actually falls counter to the notion of serendipity. Early evidence indicates strong business models bolster successful APIs.[2] Specifically, the approach of releasing free APIs to judge value doesn't always yield strong API adoption, and may require multiple iterations of APIs. Observations have shown that successful external APIs are first frequently used both internally and by strategic partners.[3,9] That is, successful APIs are designed according to a use case that has already demonstrated value to a proven business function as well as being of high quality. Hence, throwing the APIs into the wild, without continuously improving them, is ripe for failure. Success of APIs depends on the level of digital maturity of the enterprise and the corresponding API adoption models, which vary in complexity and their required investment.

For example, some enterprises might just start with API experimentation, before exposing own capabilities or reinventing them. The final, most challenging step is to embrace new business models through APIs.[4] Therefore, the businesses that want to leverage the power of APIs need reiterate both on APIs and the underlying business models in order to ensure proven value over the time.

> *The greatest risk for enterprises remains a lack of a sound API strategy.*

Abundance of API choice. API consumers, especially those developing enterprise applications, need to quickly identify APIs that satisfy their functional and non-functional requirements, such as performance capabilities and security properties. In SOA, Universal Description, Discovery and Integration (UDDI) registries[d] were meant to assist here, but they had several drawbacks. A UDDI registry was a passive entity meant to be looked up at the time of service invocation. The service information contained therein would often become stale as services went down or changed their interface and characteristics. Further, it was tied to specific technologies, such as SOAP. One way to address the abundance of choice is via automated API brokers that enable quick and easy selection of APIs that meet consumers' requirements. API brokers can be viewed as an emerging field of cognitive systems that can make these selection assessments, based on the dynamic insights about the ever-changing API ecosystem.[11] Gradually, API Broker recommendations will improve, taking into an account factors such as ease of integration and guidelines from consumer profiles.

API exposure. Mainstream adoption of APIs relies on the providers, as they encapsulate information and capabilities that will remain the bedrock of an API ecosystem. The provider must consider how it will enforce security, integrity, and quality of its exposed resources in such a diffused network of API providers and consumers. For example, given the growing need for predictive analysis for performance, where does monitoring start and who is accessing and controlling it? Moreover, how does one identify what usage trends are indicative of strains on the providers' resources? The provider has the added challenge of determining the risks involved and the investments required to mitigate them. The reality is that years of acquisitions, mergers, and crumbling knowledge of legacy IT systems have made the sheer possibility of exposing potentially valuable capabilities via APIs a complex undertaking. Any required infrastructure modernization by API provider must be balanced with incremental API exposure by carefully selecting the resources, understanding their provenance, and identifying the access methods. Furthermore, open use of APIs brings challenges of possible unintended consequences, which may be caused by, for example, the interplay between providers, consumers, who may be afraid of vendor lock-in, third-party agents, and the end users. Similarly, giving API consumers the means for evaluating API terms of service and tracking their change presents yet another barrier for enterprises. API

management gateways[e] are first steps at addressing these hurdles and help providers mitigate risk by throttling and protecting access to the backend resources.

> *Enterprises wishing to establish or be part of an API ecosystem need to clear a number of challenges.*

API consumability. Serendipitous reuse of APIs, coupled with potential "network effects" bringing spiked API usage, creates unexpected workloads that require the provider to consider elastic cloud architectures to maintain a consistent quality of service. It also creates new challenges that call for the development of metrics, which will evaluate the API output quality with respect to the ease of serendipitous reuse. An example is testing if the data exposed by the API follow the principles of linked data[1] so that they can be interlinked and become more useful. As a result, building a better semantic understanding of each API, its use, and attributes can improve consumability. As the ecosystem grows, it is important to nurture it and evolve the APIs in a manner that is consistent with the use that is observed, maintaining compatibility and optimizing for those scenarios that offer sound business value.

Overcoming the API Hype: Enabling Enterprise-Level API Ecosystems

Enterprises wishing to establish or be part of an API ecosystem need to clear a number of challenges. Firstly, they need to thrive in the flood of APIs, where the key will be to build a core set of competitively differentiated APIs. A capability that provides personalization of the API consumption experience will be vital to reach the target audience and business need. In parallel, it is critical to create an engaging ecosystem experience through consumable APIs and the data they manipulate, an environment that fosters partners to exploit the data via the APIs to create new offerings and makes users to keep coming back for more; hence, developing a "stickiness" for the ecosystem. Ecosystem members are looking for trusted providers who offer quality, reliable, long-lasting, and supported APIs. A study of API usage with open data APIs for civic apps[6] found that while there was a general lack of success with the first generation of applications, there is more traction now with efforts to improve data quality for stickiness, and efforts to increase API awareness by engaging application developers in cities and providing sustained financial incentives.

Conclusion

Success lays ahead for providers with sound API strategies that iterate through the business models concurrently improving the technical aspects of the APIs. They need to maneuver the challenges of legacy transformation and IT investments while they attract and retain consumers that drive real-time composition of their APIs, and fuel the growth of their ecosystem, successfully navigating around the hype surrounding APIs.

References

1. Bizer C., Heath T., and Berners-Lee T. Linked data—The story so far. *International Journal on Semantic Web and Information Systems* (2009), 1–22.
2. Boyd, M. Real-World API Business Models that Worked; http://bit.ly/1ZgfOwd
3. Freedman, C. 5 Things to Understand About Successful 'Open' APIs; http://bit.ly/1k6X6XX
4. Holley, K. et al. The Power of the API Economy: Stimulate Innovation, Increase Productivity, Develop New Channels and Reach New Markets. IBM Redbook; http://ibm.co/1QRltGa
5. Jacobson D., Brail G., and Woods D. *APIs: A Strategy Guide: Creating Channels with Application Programming Interfaces.* O'Reilly Media, 2011.
6. Lee M., Almirall, E., and Wareham, J. Open Data & Civic Apps: 1st Generation Failures —2nd Generation Improvements. ESADE Business School Research Paper No. 256, (Oct. 2014).
7. Pautasso, C., Zimmermann, O., and Leymann, F. RESTFul web services vs. "big" web services: Making the right architectural decision. In *Proceedings of the 17th International Conference on World Wide Web (WWW).* (ACM, 2008).
8. Salesforce Architecture. How they Handle 1.3 Billion Transactions a Day; http://bit.ly/1mvauQp
9. Takeuchi J. 8 Tips to Cultivating a Successful API Program; http://bit.ly/1OB8F2a
10. Vinoski, S. Serendipitous reuse. *IEEE Internet Computing 12*, 1 (Jan. 2008), 84–87.
11. Wittern, E. et al. A graph-based data model for API ecosystem insights. In *Proceedings of the 21st IEEE International Conference on Web Services (ICWS)*, (Anchorage, AK, 2014).

Authors

Maja Vukovic (maja@us.ibm.com) is a Research Staff Member, Master Inventor, and Member of the IBM Academy of Technology at the IBM T.J. Watson Research Center, Yorktown Heights, NY.

Jim Laredo (laredoj@us.ibm.com) is a Senior Technical Staff Member, Master Inventor, and Member of the IBM Academy of Technology at the IBM T.J. Watson Research Center, Yorktown Heights, NY.

Vinod Muthusamy (vmuthus@us.ibm.com) is a Research Staff Member at the IBM T.J. Watson Research Center, Yorktown Heights, NY.

Aleksander Slominski (aslom@us.ibm.com) is a Research Staff Member at the IBM T.J. Watson Research Center, Yorktown Heights, NY.

Roman Vaculin (vaculin@us.ibm.com) is a Research Staff Member at the IBM T.J. Watson Research Center, Yorktown Heights, NY.

Wei Tan (wtan@us.ibm.com) is a Research Staff Member at the IBM T.J. Watson Research Center, Yorktown Heights, NY.

Vijay Naik (vknaik@us.ibm.com) is a Research Staff Member at the IBM T.J. Watson Research Center, Yorktown Heights, NY.

Ignacio Silva-Lepe (isilval@us.ibm.com) is a Research Staff Member with IBM Watson Health, Yorktown Heights, NY.

Arun Kumar (kkarun@in.ibm.com) is a Senior Researcher and Research Manager, and Member of the IBM Academy of Technology, IBM Research-India.

Biplav Srivastava (sbiplav@in.ibm.com) is a Senior Researcher with IBM Research-India.

Joel W. Branch (joel.branch@espn.com) is a Manager and a Staff Software Engineer with the ESPN Advanced Technology Group (this work was done while he was a member of IBM Research).

Footnotes

a. Docker: Build, Ship and Run Any App, Anywhere; https://www.docker.com

b. http://www.twilio.com

c. http://www.stripe.com

d. UDDI Reference; http://bit.ly/1RtTL3H

e. IBM API Management; http://ibm.co/1rl5iaw